COMPLEXITY THEORY AND BOUNDED ARITHMETIC FOR TRULY FEASIBLE COMPUTATION

Kuroda, Satoru

数理解析研究所講究録
1096 巻 1999 年 35-50

COMPLEXITY THEORY AND BOUNDED ARITHMETIC FOR TRULY FEASIBLE COMPUTATION

SATORU KURODA (黒田 覚)
TOYOTA NATIONAL COLLEGE OF TECHNOLOGY,
2-1, EISEI-CHO, TOYOTA, 471-8525 JAPAN.

CONTENTS

1. Introduction

In this note we will give a survey of complexity theory and bounded arithmetic for computations within polynomial time. Nowadays, complexity theory has a lot of branches and it is almost impossible to cover all of them in this survey, so we will concentrate on the following topics.

(1) Basic notions and results of AC/NC hierarchy and other circuit classes.
(2) Recursion theoretic characterization of complexity classes
(3) Bounded arithmetic for classes between constant depth and logarithmic depth circuits

The computation below PTIME are often called truly feasible, and most works in this area are done extensibly on classes $AC^0 \subset TC^0 \subseteq NC^1 \subseteq L \subseteq NL \subseteq AC^1$ and related classes. These classes are located on the lowest level of the AC/NC hierarchy. Despite a lot of efforts, none of these classes are known to be distinct (though widely believed so) except that the lowest inclusion is proper. These two classes are separated by the parity function and the result, which is due to Furst Saxe and Sipser [9], is one of the most important result in the complexity theory.

2. Overview of Complexity Classes below P

2.1. Definitions and basic notions. First we give basic notions. We treat functions and sets of both natural numbers and binary strings. Numbers are often identified with binary strings by considering their binary expansions and conversely, binary strings are identified with corresponding natural numbers. The set of binary strings is denoted by $\{0,1\}^+$ and binary strings with length n by $\{0,1\}^n$. For a natural number x let $|x|$ be its length in binary. For any complexity class C we mean a class of functions and sets (predicates) are identified with their characteristic functions.

A circuit is a directed acyclic graph with each node labeled by either $x_1, x_2, \dots, x_n$, $\wedge$, $\vee$, $\neg$. Internal nodes are called gates and labeled by either $\wedge$, $\vee$, $\neg$. Nodes without input edges are called input and labeled by one of $x_1, x_2, \dots, x_n$. The size of a circuit is the number of gates and the depth is the length of the longest path in it. The fan-in of a gate is the number of input edges and the fan-in of the circuit is the maximum of fan-in of gates in it.

We assume that every circuit has only one output so that it computes a predicate. We say that a circuit family $C_1, \dots, C_m$ computes a function $f : \{0,1\}^n \to \{0,1\}^m$ if its bitgraph is computed by each circuit C_i. Or equivalently, putting all C_i's altogether yields a multi-output circuit that computes f. Hence we can assume that any finite function $f : \{0,1\}^n \to \{0,1\}^m$ is computed by a single circuit.

Definition 2.1. *A function $f : \{0,1\}^+ \to \{0,1\}^+$ is computed by a circuit family $\{C_n\}_{n\in\omega}$ if for all $n \in \omega$, $f\lceil_n$(f restricted to the set $\{0,1\}^n$) is computed by C_n.*

Definition 2.2. *Let $i \geq 0$. AC^i is the set of functions which are computed by some circuit family of $O((\log n)^i)$ depth, $n^{O(1)}$ circuit of unbounded fan-in. NC^i is defined in the same way except that fan-in is limited to 2.*

The following is readily proved.

Proposition 2.1. *For all $i \geq 0$, $NC^i \subseteq AC^i \subseteq NC^{i+1}$.*

Proof. The first inclusion is trivial. For the second one note that unbounded fan-in and (or) gates with n inputs can be simulated by a fan-in 2 circuit with depth $\log n$. □

The above definition of circuits, however, brings us to an unwanted situation, namely, there exists a predicate in AC^0 which is non-recursive. This is seen as follows: let $A \subset \omega$ be a non-recursive set and define a function $f : \{0,1\}^+ \to \{0,1\}^+$ by

$$f(x) = 1 \Leftrightarrow |x| \in A.$$

Then each $f\lceil_n$ is computed by either 0 or 1. But f is non-recursive since otherwise A can be decided using the algorithm for f.

To avoid such a situation we introduce a notion of uniformity.

Definition 2.3. *Let $\{C_n\}_{n\in\omega}$ be a circuit family. Direct Connection Language (DCL) of $\{C_n\}_{n\in\omega}$ is the set*

$$\{(a, b, l, 0^n) : a \text{ is the parent of } b \text{ in } C_n \text{ and } l \text{ is the label of } a\}.$$

$\{C_n\}_{n\in\omega}$ is U_{E^*}-uniform if its DCL is in DLOGTIME.

Intuitively, a circuit family is U_{E^*}-uniform if there exists a DLOGTIME algorithm that given a circuit C, determines whether C is in the circuit family. In the following, we assume that all circuit classes are U_{E^*}-uniform.

There are various versions of uniformity, e.g. logspace uniformity, P-uniformity and so on. Further discussion on this matter can be found in Johnson [16].

As stated in the introduction the lowest level of inclusion in AC/NC hierarchy is proper:

Theorem 2.2 (Furst,Saxe and Sipser [9]**).** *Parity $\notin AC^0$. Hence $AC^0 \subset NC^1$.*

Chandra, Stockmeyer and Vishkin [3] introduced the notion of constant depth reducibility and classified various functions under this reduction.

Definition 2.4. *Let f and g be functions. f is AC^0 reducible to g ($f \leq_{AC^0} g$) if there exists a AC^0 circuit family with additional gates computing g that computes f.*

$$f \equiv_{AC^0} g \Leftrightarrow f \leq_{AC^0} g \wedge g \leq_{AC^0} f.$$

Definition 2.5.

$$\begin{aligned} BinaryCount(x_1, \ldots, x_n) &= x_1 + \cdots x_n \\ Threshold^k_n(x_1, \ldots, x_n) &= 1 \textit{ iff } x_1 + \cdots x_n \geq k \end{aligned}$$

Theorem 2.3 (Chandra, Stockmeyer and Vishkin [3]**).**

$$Parity \leq_{AC^0} BinaryCount \equiv_{AC^0} Threshold \equiv_{AC^0} Multiplication.$$

The latter three functions give a characterization of an important class.

Definition 2.6. *TC^0 is the class of functions which are computable by some constant depth polynomial size circuits with additional threshold gates.*

Corollary 2.4. *BinaryCount, Threshold and Multiplication are complete for TC^0 under AC^0 reduction.*

We will concentrate on classes between AC^0 and AC^1.

2.2. Some results on logspace classes.

Definition 2.7. *Let L (resp. NL) be the class of functions which are computed by some logarithmic space bounded deterministic (resp. nondeterministic) Turing machine.*

Remark 2.1. *A function is computed by a nondeterministic Turing machine if its bitgraph is computed by the machine.*

Proposition 2.5. $NC^1 \subseteq L \subseteq NL \subseteq AC^1$.

We will state two theorems on the classes which was defined above (and also some relating classes). The first one is by N. Immerman.

Theorem 2.6 (Immerman [10]). *NL is closed under complement; i.e. if $A \in NL$ then $A^c \in NL$.*

So $co\text{-}NL = NL$ and even the logarithmic hierarchy collapses to NL.

To state the second result, we give some additional definitions. Let RL be the class of functions computed by some logspace bounded probabilistic Turing machine. SC is the class of functions which are computed by a $n^{O(1)}$ time and $(\log n)^{O(1)}$ space bounded Turing machine. Then N. Nisan showed

Theorem 2.7 (Nisan [21]). $RL \subseteq SC$.

2.3. A new hierarchy inside logarithmic depth. In this subsection, we provide a framework for the investigation of the fine structure of computations between AC^0 and AC^1 by considering circuits with $\log^{(i)} n = \log(\cdots(\log n))$ depth.

Define the iterated logarithmic function $\log^{(i)} n$ by $\log^{(1)} n = \log n$ and $\log^{(i+1)} n = \log(\log^{(i)} n)$.

Definition 2.8. *For $i \geq 1$. LD^i is the class of functions which are computable by $n^{O(1)}$ size, $(\log^{(i+1)} n)^{O(1)}$ depth unbounded fan-in circuits. MD^i is defined as LD^i but with the additional threshold gates.*

We can also define similar classes using fan-in 2 gates. However, in defining these classes, we should be more careful since merely replacing unbounded fan-in with fan-in two would yield classes which do not (known to) include AC^0. Hence we avoid such an inconvenience by defining as follows:

Definition 2.9. *ND^i is the class of functions defined as LD^i but with the additional assumption that every path from input to output contains only constantly many unbounded fan-in gates (and other gates are all fan-in two).*

We have a natural analogy between AC^i/NC^i and LD^i/ND^i, however it might not be the case (or at least hard to show) that $LD^i \subseteq ND^j$ or $ND^i \subseteq LD^j$ for any $i, j \in \omega$.

By the definition the following inclusions trivially holds.

Proposition 2.8. *For all $i \geq 1$,*

(1) $LD^i \subseteq MD^i$
(2) $AC^0 \subseteq LD^i, ND^i \subseteq AC^1$ *and* $AC^0 \subset MD^i \subseteq AC^2$.

Proof. (1) Trivial.

(2) The first one is trivial. For the second one, Note that threshold gates can be realized by NC^1 circuits. This implies $MD^i \subseteq AC^2$. $AC^0 \subset MD^i$ is implied by the fact that the parity function is not in AC^0 (cf. Furst Saxe and Sipser [9].)

□

Remark 2.2. *Since $AC^0 \neq AC^1$, either $AC^0 \neq LD^i$ or $LD^i \neq AC^1$ holds for some $i \in \omega$. The same thing also holds for the class ND^i.*

Immerman [11] showed the following alternative characterization of circuit classes which is readily applied to our case:

Definition 2.10. *A Concurrent Random Access Machine (CRAM) is a parallel machine model which has processors each of which has a local memory. CRAM also has a global memory which can be accessed from any processors. There are several methods in writing to the global memory in order to avoid write conflicts. Here we choose the PRIORITY model: there is a linear ordering on the processors, and the minimum numbered processor writes its value in a concurrent write.*

There are two sources to measure the complexity of CRAMs, time and number of processors. In the following we treat only CRAMs with polynomially number of processors. Let

$$CRAM[t(n)] = \{A \subseteq \{0,1\}^+; A \text{ is determined by some } CRAM \text{ with time } t(n)\}.$$

Theorem 2.9 (Immerman). *For all polynomially bounded and first order constructible $t(n)$,*

$$CRAM[t(n)] = AC[t(n)].$$

Corollary 2.10. *For $i \geq 1$, $LD^i = CRAM[(\log^{(i+1)} n)^{O(1)}]$.*

On the other hand, the class ND^i is characterized using the following modification of alternating Turing machines.

Definition 2.11. *An oracle alternating Turing machine (OATM) M is the alternating TM which has three kinds of states: universal, existential and query states, and has an additional oracle tape. The behavior of M is just as that of ATM in either universal or existential states. On query states M asks query to an oracle on the string which is written on the oracle tape.*

The computation of an OATM is expressed as a tree. A computation of an OATM is a path in its computation tree.

Let $Q(n)$, $S(n)$ and $T(n)$ be functions and let $\mathfrak{C}$ be some complexity class.

$OATM[S(n), T(n), Q(n), \mathfrak{C}]$ is the class of functions which are computed by some OATM M with time $T(n)$, space $S(n)$ and in each computation asks queries at most $Q(n)$ times to some oracle $A \in \mathfrak{C}$.

The following is proved in a similar manner as in Ruzzo [23].

Theorem 2.11 (Kuroda). *For $i \geq 1$,*

$$ND^i = OATM[O(\log n), (\log^{i+1} n)^{O(1)}, O(1), AC^0].$$

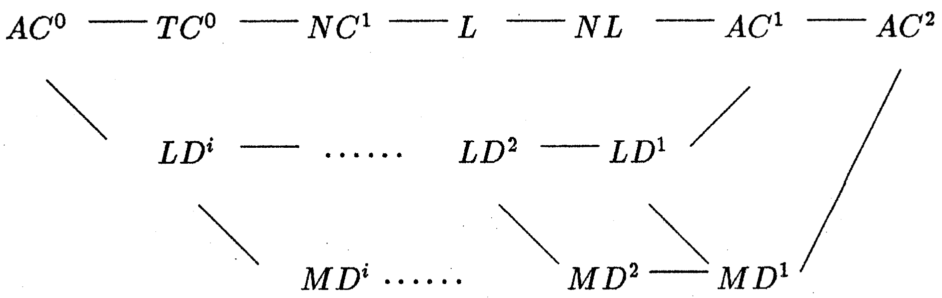

FIGURE 1. Hierarchy inside logarithmic depth

3. Weak Recursion and Complexity

A. Cobham characterized the class P using a weak form of recursion scheme called bounded recursion on notation (cf. [22]). This characterization (a.k.a. function algebra)turned out to be useful in defining a formal proof system whose derivations corresponds to polynomial time computations when S. Cook [7] defined the equational system *PV* by utilizing it. Afterward, this issue became one of the main areas in complexity theory. Among all, P. Clote studied extensively on this subject and gave characterizations for classes such as AC^0, AC^i, NC^i and so on.

3.1. Definitions and known results. In general, a function algebra are define as the closure of small number of functions (initial functions) over several functional operations which produce new functions from the previously defined ones. To illustrate this let us first recall the definition of primitive recursive functions. That is, a function is primitive recursive if it is in the smallest class containing $Z(x) = 0$, $S(x) = x + 1$, $P_n^k(x_1, \ldots, x_n) = x_k$ and closed under composition and the following primitive recursion scheme:

$$\begin{aligned} f(0, \vec{y}) &= g(\vec{y}) \\ f(x+1, \vec{y}) &= h(x, \vec{y}, f(x, \vec{y})). \end{aligned}$$

The choice of initial functions, as well as recursion schemes, varies according to the class in concern. To define weak classes below P, we need to add more functions since the recursion scheme we take is much weaker than primitive recursion. Throughout the note we shall use the following initial functions:

Definition 3.1. *INITIAL is the set of functions which consists of:*

$$Z(x) = 0, P_k^n(x_1, \dots, x_n) = x_k, s_0(x) = 2x, s_1(x) = 2x + 1,$$
$$|x| = \lceil \log_2(x+1) \rceil, Bit(x, i) = \lfloor x/2^i \rfloor mod 2, x\#y = 2^{|x| \cdot |y|}.$$

Some of these functions are unnatural as a number-theoretic functions. Nevertheless, these functions seems more natural if we identify numbers with those binary representations. For example, s_0 and s_1 are the operation of concatenation of 0 or 1 to x.

The definition of PTIME functions by Cobham is as follows:

Definition 3.2. *A function f is defined by bounded recursion on notation (BRN) from g, h_0, h_1 and k if*

$$\begin{aligned} f(0, \vec{y}) &= g(\vec{y}), \\ f(2x, \vec{y}) &= h_0(x, \vec{y}, f(x, \vec{y})), \ \ \text{if } x \neq 0 \\ f(2x+1, \vec{y}) &= h_1(x, \vec{y}, f(x, \vec{y})), \end{aligned}$$

provided that $f(x, \vec{y}) \leq k(x, \vec{y})$ for all $x, \vec{y}$.

Theorem 3.1 (Cobham). *The class of polynomial time computable functions are the smallest class containing INITIAL and closed under composition and BRN operations.*

let us turn to weaker classes. To begin with, we state the function algebra for AC^0.

Definition 3.3. *A function f is defined by concatenation recursion on notation (CRN) from g, h_0, h_1 if*

$$\begin{aligned} f(0, \vec{y}) &= g(\vec{y}), \\ h(2x, \vec{y}) &= s_{h_0(x,\vec{y})}(f(x, \vec{y})), \ \ \text{if } x \neq 0, \\ h(2x+1, \vec{y}) &= s_{h_1(x,\vec{y})}(f(x, \vec{y})). \end{aligned}$$

Theorem 3.2 (Clote [4]). *AC^0 is the smallest class containing INITIAL and closed under composition and CRN operations.*

Combining Corollary 2.4 and Theorem 3.2 we also obtain the characterization for TC^0.

Corollary 3.3. *TC^0 is the smallest class containing INITIAL and multiplication and closed under composition and CRN operations.*

J. Johannsen [14] gave a function algebra for Constable's class K based on Theorem 3.2 and proof theoretical argument which will be discussed in the next section.

Definition 3.4. *A function f is defined by weak sum (resp. product) if*

$$f(x,\vec{y}) = \sum_{i=0}^{|x|} g(i,\vec{y})(\ resp\ \prod_{i=0}^{|x|} g(i,\vec{y})).$$

Definition 3.5 (Constable). *The class K is the smallest class of functions containing INITIAL, addition, subtraction and multiplication and closed under composition, weak sum and weak product.*

Theorem 3.4 (Johannsen). *The class K is the smallest class containing INITIAL, multiplication and integer division and closed under composition and CRN operations.*

The computational complexity of the class K was quite unknown, and Theorem 3.4 revealed it to some extent as integer division is in NC^2.

Corollary 3.5. $K \subseteq NC^2$.

Various other complexity class are characterized in a similar way. For further discussion the reader should refer to an excellent survey by Clote [5].

3.2. Characterization of LD^i. Now let us define the class LD^i in a recursion theoretic manner. Let $|x|_i$ be defined as $|x|_1 = |x|$ and $|x|_{i+1} = ||x|_i|$.

Definition 3.6. *Let $i \in \omega$. A function f is defined by i-Weak Bounded Recursion on Notation (W^iBRN) from g, h_0, h_1 and k if*

$$\begin{aligned} F(0,\bar{y}) &= g(\bar{y}), \\ F(s_0(x),\bar{y}) &= h_0(x,\bar{y},f(x,\bar{y})), \text{ if } x \neq 0 \\ F(s_1(x),\bar{y}) &= h_1(x,\bar{y},f(x,\bar{y})) \\ f(x,\bar{y}) &= F(|x|_i,\bar{y}), \end{aligned}$$

provided that $F(x,\bar{y}) \leq k(x,\bar{y})$ for all $x,\bar{y}$. We call $k(x,\bar{y})$ the bounding term of the W^iBRN operation.

Theorem 3.6 (Kuroda). *For $i \geq 1$, LD^i is the smallest class of functions containing INITIAL and closed under composition, CRN and $W^{i+1}BRN$ operations.*

Proof. Let K be the closure of INITIAL under composition, CRN and $W^{i+1}BRN$. To show that $K \subseteq LD^i$, it suffices to show that LD^i is closed under $W^{i+1}BRN$ since other cases are identical to the proof of Clote and Takeuti's result stating that AC^0 is the closure of INITIAL under composition and CRN. By Corollary 2.10 we shall show that $CRAM[\log^{(i+1)} n]$ is closed $W^{i+1}BRN$. Let f be defined by $W^{i+1}BRN$ from g, h_0, h_1 and k which are computable by some CRAM's in time $(\log^{(i+1)} n)^{l_g}$, $(\log^{(i+1)} n)^{l_{g_0}}$, $(\log^{(i+1)} n)^{l_{h_1}}$ and $(\log^{(i+1)} n)^{l_k}$, respectively. On input x, the CRAM M for f computes as follows: in stage t simulate h_0 or h_1 according to the tth bit of $|x|_{i+1}$ and finally simulate g. By the inductive hypothesis each step requires at most $(\log^{(i+1)} n)^l$ steps where $l = \max l_g, l_{h_0}, l_{h_1}$, so M also terminates in $(\log^{(i+1)} n)^{l+1}$. It is also easy to see that the number of processors required by M is polynomial in $|x|$.

For the opposite direction we shall give a proof that utilizes a direct construction of LD^i circuits by weak recursion operations. Let C_n be a circuit family which computes a set $A \in LD^i$ of binary strings. (We assume the usual convention that a set $A \subseteq \{0,1\}^+$ is identified with its characteristic function.) Then C_n has size $n^{O(1)}$ and depth $\left(\log^{(i+1)} n\right)^k$ for some $k \in \omega$. Let $p(n)$ be the polynomial which bounds the number of gates in C_n. We proceed by induction on k. First let $k = 1$. By choosing a suitable encoding it is straightforward to see that the following functions are in AC^0:

$$\begin{aligned} \mathit{EncodeInput}(x) &= \text{code of the input bit } x \in \{0,1\}^+ \\ \mathit{Eval}^j_C(x) &= \text{code of the output of the } (j+1)\text{-th level of } C \\ &\quad \text{resulting from the application of } x \text{ to the gates} \\ &\quad \text{in the } i\text{-th level of } C, \\ &\quad \text{if } x \text{ is a valid code of an output from the } i\text{-th level} \end{aligned}$$

Now, starting from $\mathit{EncodeInput}(x)$ and iterating $\log^{(i+1)} n$ times the evaluation of the function Eval_C, we obtain the output of C_n on input x. This iteration procedure can be expressed by $W^{i+1}BRN$ operation since each level of output cannot exceed $p(n)$ and hence the bounding term of $W^{i+1}BRN$ is of the form $|t^{p(n)}|$ for some term t.

If $k \geq 2$, then by the induction hypothesis depth $(\log^{(i+1)} n)^{k-1}$ sub-circuits of C_n can be evaluated by functions in K. Furthermore, gathering these outputs can be done by some AC^0 function. So applying $W^{i+1}BRN$ one more time yields the output of C_n. □

Corollary 3.7. *MD^i is the smallest class containing INITIAL and multiplication and closed under composition CRN and $W^{i+1}BRN$.*

4. Bounded Arithmetic for Weak Complexity Classes

Bounded arithmetic theories for complexity classes below P were first defined by Clote and Takeuti [6], and Allen [1]. Recently, J. Johannsen [?], and C. Pollet [15] studied such theories for TC^0 and the author [18], [?] studies theories for classes below NC^1. Here we survey the latter two results.

First let us give basic notions on bounded arithmetic. The language of bounded arithmetic $\mathfrak{L}_1$ consists of function symbols, $Z(x) = 0$, $P^n_k(x_1, \dots, x_n) = x_k$, $s_0(x) = 2x$, $s_1(x) = 2x+1$, $|x| = \lceil \log_2(x+1) \rceil$, $x \# y = 2^{|x|\cdot|y|}$, and $Bit(x,i) = \lfloor x/2^i \rfloor \mod 2$ and a predicate symbol $\leq$.

A quantifier is called *bounded* if it is either of the form $\forall x \leq t$ or $\exists x \leq t$ and *sharply bounded* if it is either of the form $\forall x \leq |t|$ or $\exists x \leq |t|$. A formula is bounded if all quantifiers are bounded and sharply bounded if all quantifiers are sharply bounded. Σ^b_0 is the set of sharply bounded formulae. Σ^b_1 is the set of formulae in which all non-sharply bounded quantifiers are positive appearances of existential quantifiers. Π^b_1 is defined in the same way by replacing existential to universal. Σ^b_i and Π^b_i $(i \geq 2)$ are define in an analogous manner.

BASIC is a finite set of axioms which define symbols in $\mathfrak{L}_1$. Let Φ be a set of formulae.

- Φ-Bit-Comprehension:

$$\exists y < 2^{|t|} \forall i < |t| \, [Bit(i, x) = 1 \leftrightarrow \varphi(i)],$$

- Φ-replacement:

$$\begin{aligned}&\forall x \leq |s| \exists y \leq t(x) \varphi(x, y)\\ &\to \exists w < SqBd(s, t(|s|)) \forall x \leq |s| \, [\beta(w, x+1) \leq t(x) \wedge \varphi(x, \beta(w, x+1))],\end{aligned}$$

- Φ-LIND:

$$\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(|x|),$$

- Φ-$L^i IND$:

$$\varphi(0) \wedge \forall x(\varphi(x) \to \varphi(x+1)) \to \forall x \varphi(|x|_i),$$

where $\varphi \in \Phi$.

We shall be interested in a provably total functions of a given bounded arithmetic theory whose defining formula has some specific logical complexity.

Definition 4.1. *Let T be a theory of bounded arithmetic. A function f is Σ_i^b definable in T if there exists a formula $\varphi \in \Sigma_i^b$ such that*

$$\begin{aligned}&T \vdash \forall x \exists y \varphi(x, y),\\ &T \vdash \forall x, y, z(\varphi(x, z) \wedge \varphi(y, z) \to x = y),\\ &N \models \forall x \varphi(x, f(x)).\end{aligned}$$

4.1. Some weak theories for circuit classes. Our weakest theory should be that for the class AC^0. Such theories are defined by Clote and Takeuti [ ], F. Ferreira [ ], and the author. Here we choose the one by the author.

Let $\mathfrak{L}_{AC}$ be the language which consists of symbols in $\mathfrak{L}_1$ plus function symbols for each AC^0 functions.

Definition 4.2. *AC^0CA is the $\mathfrak{L}_{AC}$ theory which consists of the following axioms:*

- *defining axioms for all $f \in AC^0$ given by the recursion theoretic characterization of AC^0 (Theorem 3.2).*
- *Σ_0^b-LIND.*

Theorem 4.1. *A function f is in AC^0 if and only if it is Σ_0^b definable in AC^0CA.*

Here we shall present a model theoretical proof of Theorem 4.1. First we use the following fact by Łos and Tarski.

Lemma 4.2. *A theory T is Π_0^1 axiomatizable if and only if it is preserved under substructures, i.e. if $M \models T$ and N is a substructure of M then $N \models T$.*

Then using a witnessing argument in an arbitrary model of AC^0CA we conclude that AC^0CA is preserved under substructures. Hence we have that

Lemma 4.3. *AC^0CA is Π_0^1 axiomatizable.*

Now recall Herbrand's theorem for Π_0^1 axiomatizable theories.

Theorem 4.4 (Herbrand). *Let T be a Π_0^1 axiomatizable theory and suppose that $T \vdash \forall x \exists y \varphi(x, y)$ for an open formula φ. Then there exists a term t such that $T \vdash \forall x \varphi(x, f(x))$.*

Again by a witnessing argument it is shown that any Σ_0^b formula is equivalent to some open formula in AC^0CA. So Theorem 4.1 is proved.

Johannsen studied systems for the class TC^0 and related classes. Here we survey his result (partly joint work with C. Pollet) without proofs.

Definition 4.3. *The Δ_1^b-comprehension rule, Δ_1^b-COMP, is the following inference rule*

$$\frac{\varphi(x) \leftrightarrow \psi(x)}{COMP_{\varphi}(t),}$$

where $COMP_A$ is the bit comprehension for φ, $\varphi \in \Sigma_1^b$, $\psi \in \Pi_1^b$ and t is an arbitrary term.

Definition 4.4. *Let Δ_1^b-CR be the theory whose axioms are BASIC, LIND for open formulae and Δ_1^b-COMP rule.*

Theorem 4.5 (Johannsen and Pollet). *The Σ_1^b definable functions of Δ_1^b-CR are precisely TC^0.*

They also used so called KPT witnessing theorem to show

Theorem 4.6 (Johannsen and Pollet). *if $S_2^i = \Delta_1^b$-CR then NP is contained in nonuniform TC^0.*

Johannsen found that Δ_1^b-CR extended by a single function exactly defines Constable's class K.

Definition 4.5. *Integer division $\lfloor \frac{x}{y} \rfloor$ is define by*

$$\lfloor \tfrac{x}{0} \rfloor = 0,$$
$$y > 0 \to y \cdot \lfloor \tfrac{x}{y} \rfloor \leq x < y \cdot \lfloor \tfrac{x}{y} \rfloor + y.$$

The theory Δ_1^b-CR[div] is the theory Δ_1^b-CR extended by the function integer division.

Theorem 4.7 (Johannsen). *Constable's class K is exactly the class of functions which are Σ_1^b definable in Δ_1^b-CR[div].*

4.2. The theory L^i_2. Defining a formal theory for the class LD^i might be a little messy compared to other systems like S^i_2. First we shall introduce the notion of essentially sharply boundedness.

Definition 4.6. *Let T be a theory. A formula φ is esb in T if it belongs to the smallest class $\mathfrak{F}$ satisfying the following conditions:*

- *every atomic formula is in $\mathfrak{F}$.*
- *$\mathfrak{F}$ is closed under boolean connectives and sharply bounded quantifications.*
- *If $\varphi_0, \varphi_1 \in \mathfrak{F}$ and*

$$T \vdash \exists x \leq s(\vec{a})\varphi_0(\vec{a}, x)$$
$$T \vdash \forall x, y \leq s(\vec{a})(\varphi_0(\vec{a}, x) \wedge \varphi_0(\vec{a}, y) \rightarrow x = y)$$

then $\exists x \leq s(\vec{a})(\varphi_0(\vec{a}, x) \wedge \varphi_1(\vec{a}, x))$ and $\forall x \leq s(\vec{a})(\varphi_0(\vec{a}, x) \rightarrow \varphi_1(\vec{a}, x))$ are in $\mathfrak{F}$.

A formula is $ep\Sigma^b_1$ in T if it is of the form $\exists x_1 \leq t_1 \cdots \exists x_k \leq t_k \varphi(x_1, \ldots, x_k)$ where φ is esb in T.

Definition 4.7. *A function f is esb definable in a theory T if there exist an esb formula φ in T that defines f.*

The following immediately holds by the definition.

Proposition 4.8. *Let $\varphi(\vec{x}, y)$ esb-define a function f in a theory T. Then the following formulae are equivalent in $T(f)$*

- $\exists x \leq s(\vec{a})(\varphi(\vec{a}, x) \wedge \psi(\vec{a}, x))$
- $\forall x \leq s(\vec{a})(\varphi(\vec{a}, x) \rightarrow \psi(\vec{a}, x))$
- $\psi(\vec{a}, f(\vec{a}))$.

Definition 4.8. *Let φ be an esb formula in T. Then we denote the equivalent sharply bounded formula (in the extended language) by φ^{sb} (called sb version of φ). If φ is $ep\Sigma^b_1$ of the form $\exists x_1 \leq t_1 \cdots \exists x_k \leq t_k \varphi(x_1, \ldots, x_k)$ where φ is esb then φ^{sb} denotes the formula*

$$\exists x_1 \leq t_1 \cdots \exists x_k \leq t_k \varphi^{sb}(x_1, \ldots, x_k).$$

For sequents and inference rules, their sb versions are defined analogously.

Now we define a theory whose provably total functions are exactly those in LD^i.

Definition 4.9. *L^i_2 is the $\mathfrak{L}_1$ theory which consists of the following axioms:*

- *BASIC*
- *Σ^b_0-Bit-Comprehension*
- *Σ^b_0-LIND*
- *$ep\Sigma^b_1$-$L^{i+1}IND$.*

Remark 4.1. *Let $f \in AC^0$ and $L^i_2(f)$ be the theory L^i_2 extended by the function symbol f together with its defining axioms. Then $L^i_2(f)$ is a conservative extension of L^i_2. Hence we can regard $ACCA$ as a subtheory of L^i_2.*

First we show the definability of LD^i functions in L^i_2.

Theorem 4.9 (Kuroda). *If $f \in LD^i$ then f is esb definable in L^i_2.*

Proof. The proof is by induction on the complexity of $f \in AL^i$.

By the proof of Theorem 4.1, all INITIAL functions are Σ^b_0 definable in T^0AC^0, hence also in L^i_2. The same argument implies that the closure under composition and CRN are also proved within T^0AC^0. So it suffices to show that esb definable functions of L^i_2 are closed under L^iBRN operation.

Let f be define by L^iBRN from g, h_0, h_1 and k each has Σ^b_1 definition in L^i_2. Let $\Phi(x, \bar{y})$ be the formula expressing that "w is a sequence of the computation of f". Then it is readily seen that Φ is $ep\Sigma^b_1$ in L^i_2 and

$$L^i_2 \vdash \Phi(0, \bar{y}) \wedge \forall x(\Phi(x, \bar{y}) \to \Phi(x+1, \bar{y})).$$

So by $ep\Sigma^b_1$-L^iIND we have $L^i_2 \vdash \forall x \Phi(|x|_i, \bar{y})$. Hence the Σ^b_1 formula Φ defines f provably in L^i_2. □

Now we shall show the converse to the previous theorem. Namely, All Σ^b_1 consequences of L^i_2 are witnessed by some LD^i functions.

Theorem 4.10 (Kuroda). *Let $\varphi \in ep\Sigma^b_1$ be such that $L^i_2 \vdash \forall x \exists y \varphi(x, y)$. Then there exists a function $f \in LD^i$ such that $L^i_2 \vdash \forall x \varphi(x, f(x))$.*

The proof is by the witnessing method.

Theorem 4.10 is a corollary to the following theorem.

Theorem 4.11. *Let $\Gamma \to \Delta$ be provable in L^i_2 and $\Gamma^{sb} \to \Delta^{sb}$ be of the form*

$$\begin{aligned}\exists x \leq s_1 A^{sb}_1(\vec{a}, x) \wedge & \cdots \exists x \leq s_m A^{sb}_m(\vec{a}, x) \\ & \to \exists y \leq t_1 B^{sb}_1(\vec{a}, x) \wedge \cdots \exists y \leq t_m B^{sb}_n(\vec{a}, x)\end{aligned}$$

where $A_1, \ldots, A_m, B_1 \ldots, B_n$ are sharply bounded. Then there exist functions $f_1, \ldots, f_n \in LD^i$ such that

$$\begin{aligned}b_1 \leq s_1(\vec{a}) A^{sb}_1(\vec{a}, x) \wedge & \cdots b_m \leq s_m(\vec{a}) A^{sb}_m(\vec{a}, x) \\ & \to f_1(\vec{a}, \vec{b}) \leq t_1(\vec{a}) B^{sb}_1(\vec{a}, f(\vec{a}, \vec{b})) \wedge \cdots f_n(\vec{a}, \vec{b}) \leq t_m(\vec{a}) B^{sb}_n(\vec{a}, \vec{b}),\end{aligned}$$

where $\vec{b} = b_1, \ldots, b_m$.

Proof. Induction on the number of sequences in the L^i_2 proof of the sequent $\Gamma \to \Delta$. The precise proof will appear in [20]. □

4.3. KPT witnessing theorem and conditional separation. It is much more natural if we can replace $s\Sigma_1^b$-$L^{i+1}IND$ with Σ_1^b-$L^{i+1}IND$ in the definition of L_2^i. However, it is unknown whether this extended theory corresponds to the class LD^i. Nevertheless, we can show that this theory may be slightly stronger than L_2^i.

Definition 4.10. $L_2^i(\Sigma_1^b) = L_2^i + \Sigma_1^b$-$L^{i+1}IND$.

As in Johannsen and Pollet [15], we use KPT witnessing theorem to separate $L_2^i(\Sigma_1^b)$ from weaker theory AC^0CA.

Theorem 4.12. *The theory AC^0CA is Π_1^0 axiomatized.*

Therefore by Herbrand's theorem for $\forall\exists\forall\Sigma_1^b$ formula we obtain

Theorem 4.13. *Let $\varphi \in \Sigma_1^b$ and suppose $AC^0CA \vdash \exists x \forall y \varphi(a, x, y)$. Then there exists a finite number of functions $f_1, \ldots, f_k \in AC^0$ such that AC^0CA proves*

$$\varphi(a, f_1(a), b_1) \vee \varphi(a, f_1(a, b_1), b_2) \vee \cdots \vee \varphi(a, f_k(a, b_1, \ldots, b_{k-1}), b_k).$$

This witnessing theorem is known to be realized by the following Ω principle:

Either	$\forall z P(a, f_1(a), z)$	or if b_1 is such that $\neg R(a, f_1(a), b_1)$
then either	$\forall z P(a, f_2(a, b_1), z)$	or if b_2 is such that $\neg R(a, f_2(a, b_1), b_2)$
	...	
then	$\forall z P(a, f_k(a, b_1, \ldots, b_{k-1}), z)$	

For a binary predicate $R(x, y)$ define

$$R^*(x, y) \equiv R(x, y) \wedge \forall z(|x|_i \leq |z|_i < |y|_i \to \neg R(x, z)).$$

Let $\Omega^i(R)$ be the Ω principle for the optimization problem R^*. Then we have

Theorem 4.14. *$AC^0CA = L_2^i(\Sigma_1^b)$ implies the principle $\Omega^i(AC^0)$.*

Proof. The proof is essentially the same as in Johannsen and Pollet [15]. □

Also we have

Theorem 4.15. *The principle $\Omega^i(AC^0)$ implies that $nonuniformAC^0 \subseteq NP$.*

Hence as a corollary we have

Corollary 4.16. $AC^0CA \neq L_2^i(\Sigma_1^b)$.

It seems hard to show that $AC^0 \neq LD^i$. So by Corollary 4.16 it is also hard to show that $L_2^i = L_2^i(\Sigma_1^b)$. But this may be possible since Corollary 4.16 says nothing about Σ_1^b conservation between AC^0CA and $L_2^i(\Sigma_1^b)$.

5. Remarks for Future Researches

5.1. Complexity class and function algebras.

Problem 5.1. *Find a function algebra for ND^i.*

In section 3.2 we gave a recursion theoretic characterization of the class LD. But the author do not know whether the class ND^i admits similar characterization. As for function algebras for complexity classes, S. Bellantoni and S. Cook gave a new characterization using two sorts of parameters (safe and normal) and *safe recursion scheme*. The main advantages of their characterization are that it does not require the artificial function $x\#y$ and also that it eliminates the bound for growth rate in the recursion scheme. Izumi Takeuti asked whether LD^i admits safe recursion theoretic characterization.

It seems that the separation of classes LD^i's and AC^0 or AC^1 (or other classes) is very difficult. In general these separation problems become much easier if we allow oracles. So,

Problem 5.2. *Show that there exists an oracle A such that $AC^0[A] \neq LD^i[A]$, $LD^i[A] \neq AC^1[A]$ or $LD^i[A] \neq LD^j[A]$ for $i \neq j$.*

5.2. Some questions on the theory L_2^i and other related systems. It seems more likely that we can replace L_2^i by the following theory.

Definition 5.1. *$L_2^i(\Delta_1^b)$ is the theory L_2^i extended by the following Δ_1^b-$L^{i+1}IND$:*

$$\forall x(\varphi(x) \leftrightarrow \neg\psi(x)) \to L^{i+1}IND(\varphi).$$

Then the problem is

Problem 5.3. *Show that Σ_1^b consequences of $L_2^i(\Delta_1^b)$ corresponds to LD^i.*

The problem of determining the computational complexity of Σ_1^b-$L^{i+1}IND$ is also interesting. More generally we may ask

Problem 5.4. *What is the computational complexity of Σ_k^b consequences of Σ_k^b-$L^{i+1}IND$?*

The relation between L^iIND for $i \in \omega$ is also interesting:

Problem 5.5. *Does Σ_{k+1}^b-L^iIND imply Σ_k^b-L^jIND for some $i < j$?*

REFERENCES

1. B. Allen, *Arithmetizing uniform NC*, Annals of Pure and Applied Logic, 53(1), 1991, 1–50.
2. S. R. BUSS, *Bounded Arithmetic.* Bibliopolis. (1986).
3. A. K. Chandra, L. Stockmeyer and U. Vishkin, *Constant depth reducibility*, SIAM J. Compt., 13, 1984, 423–439.
4. P. Clote, Sequential machine-independent characterizations of the parallel complexity classes ALOGTIME, AC^k, NC^k and NC, in: P.J. Scott and S.R. Buss eds., *Feasible Mathematics*, (Birkhäuser, 1990) 49–70.
5. P. Clote, *Computation models and function algebras*, preprint.
6. P. CLOTE AND G. TAKEUTI, *First Order Bounded Arithmetic and Small Circuit Complexity classes.* In: Feasible Mathematics II, Birkhäuser. pp. 154-218 (1995).
7. S. Cook, *Feasibly constructive proofs and the propositional calculus*, in: Proc. 7th Annual ACM Syymp. on Theory of Computing, (1975), 83–97.
8. F. Ferreira, *On end-extensions of models of* $\neg$exp, Math. Log. Quart., 42, 1996, 1–18.
9. M. Furst, J.B. Saxe and M. Sipser, *Parity, Circuits and the Polynomial-Time Hierarchy, Mathematica Systems Theory*, (1984), 13–27.
10. N. Immerman, *Nondeterministic space is closed under complement, SIAM J. Comput.*, (1988), 935–938.
11. N. Immerman, *Expressibility and parallel complexity, SIAM J. Comput.*, (1989) 625–638.
12. N. Immerman, *Descriptive Complexity*, (Springer Verlag, New York, 1999).
13. J. JOHANNSON, *A Bounded Arithmetic Theory for Constant Depth Threshold Circuits.* In: Gödel '96, Lecture Notes in Logic 6, pp. 224-234 (1996).
14. J. Johannsen, *Weak Bounded Arithmetic, the Diffie-Hellman Problem and Constable's Class K*, preprint.
15. J. JOHANNSON AND C. POLLET, *On* Δ_1^b *Bit Comprehension Rule*, to appear in: Proceedings of Logic Colloquium '98.
16. D.S. Johnson, *A Catalog of Complexity Classes*, in: J. van Leeuwen ed., *Handbook of Theoretical Computer Science Vol. 1, Algorithms and Complexity*, (The MIT Press/Elsevier, 1990)
17. J. KRAJÍČEK, *Bounded Arithmetic, Propositional Logic, and Complexity Theory.* Cambridge Univ. Press, (1995).
18. S. KURODA, *On a Theory for* AC^0 *and the Strength of the Induction Scheme.* Math. Log. Quart., pp. 417-426 (1998).
19. S. KURODA, *Function Algebras for Very Small Depth Circuits*, submitted.
20. S. KURODA, *Bounded Arithmetic for slow growing depth circuits*, in preparation.
21. N. Nisan, $RL \subseteq SC$,
22. H. E. Rose, *Subrecursion: Functions and hierarchies*, Oxford Logic Guide 9, (1984)
23. W.L. Ruzzo, *On Uniform Circuit Complexity, J. Compt. Sys. Sci.*, 22, (1981), 365-383.